Delft University of Technology

Fac. of Information Technology and Systems **Control Systems Engineering**

Technical report CSE02-004

MPC for continuous piecewise-affine systems*

B. De Schutter and T.J.J. van den Boom

If you want to cite this report, please use the following reference instead:
B. De Schutter and T.J.J. van den Boom, "MPC for continuous piecewise-affine systems," *Systems & Control Letters*, vol. 52, no. 3–4, pp. 179–192, July 2004. doi:10.1016/j.sysconle.2003.11.010

Control Systems Engineering
Faculty of Information Technology and Systems
Delft University of Technology
Delft, The Netherlands
Current URL: https://www.dcsc.tudelft.nl

MPC for Continuous Piecewise-Affine Systems

B. De Schutter and T.J.J. van den Boom

Delft Center for Systems and Control, Delft University of Technology
Mekelweg 2, 2628 CD Delft, The Netherlands
email: {b.deschutter,t.j.j.vandenboom}@dcsc.tudelft.nl

Abstract

A large class of hybrid systems can be described by a max-min-plus-scaling (MMPS) model (i.e., using the operations maximization, minimization, addition and scalar multiplication). First, we show that continuous piecewise-affine systems are equivalent to MMPS systems. Next, we consider model predictive control (MPC) for these systems. In general, this leads to nonlinear, nonconvex optimization problems. We present a new MPC method for MMPS systems that is based on canonical forms for MMPS functions. In case the MPC constraints are linear constraints in the inputs only, this results in a sequence of linear optimization problems such that the MPC control can often be computed in a much more efficient way than by just applying nonlinear optimization as was done in previous work.

Keywords: model predictive control, hybrid systems, piecewise-affine systems, max-min-plus-scaling systems.

1 Introduction

Hybrid systems contain both analog (continuous) and logical (discrete, switching) dynamics. Typical examples are manufacturing systems, telecommunication and computer networks, traffic control systems, digital circuits, and logistic systems. Piecewise-affine (PWA) models are often used to describe the behavior of hybrid systems since they form the "simplest" extension of linear systems that can still model nonlinear and nonsmooth processes with arbitrary accuracy and since they can deal with hybrid phenomena. PWA systems have been studied by many authors [2, 4, 8, 9, 19, 20, 21, 25, 27]. In particular, Sontag has considered PWA systems from a classical control perspective [25]. He has also studied specific properties like representation, realization, observability, and decidability questions. Furthermore, recently, Morari, Bemporad, *et al.* [1, 2, 4, 14] have developed a model predictive control approach for PWA systems.

Another modeling framework for hybrid systems consists in max-min-plus-scaling (MMPS) models, which use maximization, minimization, addition and scalar multiplication. Recently, a link between *constrained* MMPS systems and PWA systems (and other classes of hybrid systems) has been established [17, 18]. In this paper we will present a direct connection between *continuous* PWA systems and MMPS systems (without the need to introduce additional auxiliary variables or extra constraints as was done in [17, 18]).

In [12] we have extended the model predictive control (MPC) framework to MMPS systems. In this paper we will use the link between continuous PWA systems and MMPS systems to present a new approach to MPC for continuous PWA systems. In order to compute an MPC controller for a continuous PWA system or an MMPS system, we have to solve a nonlinear, nonconvex optimization problem at each sample step. For the case that the MPC constraints are linear constraints in the inputs, we propose a new optimization approach that is based on canonical forms for MMPS functions and that is similar to the cutting-plane algorithm for convex optimization problems. The proposed approach consists in solving several linear programming problems and is more efficient than the algorithms used in [12], which are based on multi-start nonlinear local optimization (sequential quadratic programming) or on the extended linear complementarity problem.

This paper is organized as follows. In Section 2 we present MMPS and PWA functions and systems, and we establish the equivalence between continuous PWA systems and MMPS systems. Next, we consider canonical forms for MMPS functions. In Section 4 we briefly recapitulate our previous results on MPC for MMPS systems. Because of the link between continuous PWA and MMPS systems, this approach can also be used for continuous PWA systems. Next, we present an efficient algorithm to solve the MMPS-MPC and PWA-MPC optimization problems. We conclude with two worked examples.

2 Continuous PWA systems and MMPS systems

2.1 MMPS and PWA functions and systems

Definition 2.1 *An MMPS function $f : \mathbb{R}^n \to \mathbb{R}^m$ is defined by the recursive grammar*

$$f(x) := x_i \mid \alpha \mid \max(f_k(x), f_l(x)) \mid \min(f_k(x), f_l(x)) \mid f_k(x) + f_l(x) \mid \beta f_k(x) \quad , \tag{1}$$

with $i \in \{1,\dots,n\}$, $\alpha, \beta \in \mathbb{R}$, and where $f_k : \mathbb{R}^n \to \mathbb{R}^m$, $f_l : \mathbb{R}^n \to \mathbb{R}^m$ are again MMPS functions; the symbol $\mid$ stands for "or", and max *and* min *are performed entrywise.*

Systems that can be described by a state space model of the form

$$x(k) = \mathscr{M}_x(x(k-1), u(k)) , \quad y(k) = \mathscr{M}_y(x(k), u(k)) \quad , \tag{2}$$

with input u, output y, and state x, and where $\mathscr{M}_x, \mathscr{M}_y$ are (vector-valued) MMPS functions, are called *MMPS systems*.

Definition 2.2 [8] *A scalar-valued function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be a continuous PWA function if and only if the following conditions hold:*

1. *The domain space $\mathbb{R}^n$ is divided into a finite number of polyhedral regions $R_{(1)}, \dots, R_{(N)}$.*
2. *For each $i \in \{1,\dots,N\}$, f can be expressed as $f(x) = \alpha_{(i)}^T x + \beta_{(i)}$ for any $x \in R_{(i)}$ with $\alpha_{(i)} \in \mathbb{R}^n$ and $\beta_{(i)} \in \mathbb{R}$.*
3. *f is continuous on any boundary between two regions.*

A vector-valued function is continuous PWA if each of its components is continuous PWA.

A PWA system is a system of the form

$$x(k) = \mathscr{P}_x(x(k-1), u(k)) , \quad y(k) = \mathscr{P}_y(x(k), u(k)) \quad , \tag{3}$$

with $\mathscr{P}_x, \mathscr{P}_y$ vector-valued PWA functions. If $\mathscr{P}_x, \mathscr{P}_y$ are continuous, we say that the system is continuous PWA. Note that PWA models can also be used as approximations of more general state space models of the form $x(k) = f(x(k-1), u(k))$, $y(k) = g(x(k), u(k))$, with f, g continuous and nonlinear.

For more information on PWA functions and PWA systems we refer to [2, 8, 9, 19, 20, 21, 25] and the references therein.

2.2 Equivalence of continuous PWA and MMPS systems

Theorem 2.3 [16, 24] *If f is a continuous PWA function of the form given in Definition 2.2, then there exist index sets $I_1, \ldots, I_\ell \subseteq \{1, \ldots, N\}$ such that*

$$f = \max_{j=1,\ldots,\ell} \min_{i \in I_j} \left(\alpha_{(i)}^T x + \beta_{(i)}\right) \ .$$

From the definition of MMPS functions it follows that (see also [16, 24]):

Lemma 2.4 *Any MMPS function is also a continuous PWA function.*

From Theorem 2.3 and Lemma 2.4 it follows that continuous PWA systems and MMPS systems are equivalent, i.e., for a given continuous PWA model there exists an MMPS model (and vice versa) such that the input-output behavior of both models coincides:

Proposition 2.5 *Continuous PWA systems and MMPS systems are equivalent.*

This is an extension of the results of [17, 18], which already prove an equivalence between (not necessarily continuous) PWA models and MMPS models, but there some extra auxiliary variables and some additional algebraic MMPS constraints between the states, the inputs and the auxiliary variables were required to transform the PWA model into an MMPS model.

3 Canonical forms of MMPS functions

Now we consider some easily verifiable properties of the max and min operators that will be used in the proof of the main theorem of this paper. Let $\alpha, \beta, \gamma, \delta \in \mathbb{R}$.

- Since minimization is distributive w.r.t. maximization, i.e., $\min\big(\alpha, \max(\beta,\gamma)\big) = \max\big(\min(\alpha, \beta), \min(\alpha,\gamma)\big)$, and maximization is distributive w.r.t. minimization, we have

$$\min\big(\max(\alpha,\beta),\max(\gamma,\delta)\big) = \max\big(\min(\alpha,\gamma),\min(\alpha,\delta),\min(\beta,\gamma),\min(\beta,\delta)\big) \qquad (4)$$

$$\max\big(\min(\alpha,\beta),\min(\gamma,\delta)\big) = \min\big(\max(\alpha,\gamma),\max(\alpha,\delta),\max(\beta,\gamma),\max(\beta,\delta)\big) \ . \qquad (5)$$

- Since addition is distributive w.r.t. minimization and maximization, we have

$$\min(\alpha,\beta) + \min(\gamma,\delta) = \min(\alpha+\gamma, \alpha+\delta, \beta+\gamma, \beta+\delta) \qquad (6)$$

$$\max(\alpha,\beta) + \max(\gamma,\delta) = \max(\alpha+\gamma, \alpha+\delta, \beta+\gamma, \beta+\delta) \ . \qquad (7)$$

- The min and max operators are related as follows:

$$\max(\alpha,\beta) = -\min(-\alpha,-\beta), \qquad \min(\alpha,\beta) = -\max(-\alpha,-\beta) \ . \qquad (8)$$

- If $\rho \in \mathbb{R}$ is positive, then

$$\rho \max(\alpha,\beta) = \max(\rho\alpha, \rho\beta), \qquad \rho \min(\alpha,\beta) = \min(\rho\alpha,\rho\beta) \ . \qquad (9)$$

Theorem 3.1 *A scalar-valued MMPS function f can be rewritten into the min-max canonical form*

$$f = \min_{i=1,\ldots,K} \max_{j=1,\ldots,n_i} (\alpha_{(i,j)}^T x + \beta_{(i,j)}) \tag{10}$$

or into the max-min canonical form $f = \max_{i=1,\ldots,L} \min_{j=1,\ldots,m_i} (\gamma_{(i,j)}^T x + \delta_{(i,j)})$ for some integers K, L, $n_1, \ldots, n_K$, $m_1, \ldots, m_L$, vectors $\alpha_{(i,j)}, \gamma_{(i,j)}$, and real numbers $\beta_{(i,j)}, \delta_{(i,j)}$.
For vector-valued MMPS functions the above statements hold componentwise[1].

Proof: We will only prove the theorem for the min-max canonical form since the proof for the max-min canonical form is similar. It is easy to verify that if f_k and f_l are affine functions, then the functions that results from applying the basic constructors of an MMPS function (max, min, +, and scaling — cf. Definition 2.1) are in min-max canonical form[2]. Now we use a recursive argument that consists in showing that if we apply the basic constructors of an MMPS function to two (or more) MMPS functions in min-max canonical form, then the result can again be transformed into min-max canonical form. Consider two MMPS functions f and g in min-max canonical form[3]: $f = \min(\max(f_1, f_2), \max(f_3, f_4))$ and $g = \min(\max(g_1, g_2), \max(g_3, g_4))$. Now we show that $\max(f, g)$, $\min(f, g)$, $f + g$ and βf with $\beta \in \mathbb{R}$ can again be written in min-max canonical form:

- $$\begin{aligned} \max(f,g) &= \max\big[\min\big(\max(f_1,f_2),\max(f_3,f_4)\big),\ \min\big(\max(g_1,g_2),\max(g_3,g_4)\big)\big] \\ &= \max\big[\max\big(\min(f_1,f_3),\min(f_1,f_4),\min(f_2,f_3),\min(f_2,f_4)\big), \\ &\qquad \max\big(\min(g_1,g_3),\min(g_1,g_4),\min(g_2,g_3),\min(g_2,g_4)\big)\big] \qquad \text{(by (4))} \\ &= \max\big(\min(f_1,f_3),\min(f_1,f_4),\min(f_2,f_3),\min(f_2,f_4), \\ &\qquad \min(g_1,g_3),\min(g_1,g_4),\min(g_2,g_3),\min(g_2,g_4)\big) \\ &= \min\big(\max(f_1,f_1,f_2,f_2,g_1,g_1,g_2,g_2),\max(f_1,f_1,f_2,f_2,g_1,g_1,g_2,g_4),\ldots \\ &\qquad \max(f_3,f_4,f_3,f_4,g_3,g_4,g_3,g_4)\big) \qquad \text{(since max is distributive w.r.t. min)} \end{aligned}$$
- $$\begin{aligned} \min(f,g) &= \min\big[\min\big(\max(f_1,f_2),\max(f_3,f_4)\big),\ \min\big(\max(g_1,g_2),\max(g_3,g_4)\big)\big] \\ &= \min\big(\max(f_1,f_2),\max(f_3,f_4),\max(g_1,g_2),\max(g_3,g_4)\big) \end{aligned}$$
- $$\begin{aligned} f+g &= \min\big(\max(f_1,f_2),\max(f_3,f_4)\big) + \min\big(\max(g_1,g_2),\max(g_3,g_4)\big) \\ &= \min\big(\max(f_1,f_2)+\max(g_1,g_2),\max(f_1,f_2)+\max(g_3,g_4), \\ &\qquad \max(f_3,f_4)+\max(g_1,g_2),\max(f_3,f_4)+\max(g_3,g_4)\big) \qquad \text{(by (6))} \\ &= \min\big(\max(f_1+g_1,f_1+g_2,f_2+g_1,f_2+g_2), \\ &\qquad \max(f_1+g_3,f_1+g_4,f_2+g_3,f_2+g_4),\max(f_3+g_1,f_3+g_2,f_4+g_1,f_4+g_2), \\ &\qquad \max(f_3+g_3,f_3+g_4,f_4+g_3,f_4+g_4)\big) \qquad \text{(by (7))} \end{aligned}$$
- $$\begin{aligned} \beta f &= \beta \min\big(\max(f_1,f_2),\max(f_3,f_4)\big) \\ &\overset{\beta \geqslant 0}{=} \min\big(\max(\beta f_1,\beta f_2),\max(\beta f_3,\beta f_4)\big) \qquad \text{if } \beta \geqslant 0 \qquad \text{(by (9))} \\ &\overset{\beta<0}{=} -|\beta| \min\big(\max(f_1,f_2),\max(f_3,f_4)\big) \qquad \text{if } \beta < 0 \\ &= -\min\big(\max(|\beta| f_1,|\beta| f_2),\max(|\beta| f_3,|\beta| f_4)\big) \qquad \text{(by (9))} \end{aligned}$$

---

[1] Or alternatively, $\alpha_{(i,j)}, \gamma_{(i,j)}$ are matrices, and $\beta_{(i,j)}, \delta_{(i,j)}$ are vectors.

[2] We allow "void" min or max statements of the form $\min(s)$ or $\max(s)$, which by definition are equal to s for any expression s. Alternatively, we can write $\min(s,s)$ or $\max(s,s)$.

[3] For the sake of simplicity we only consider two min-terms in f and g, each of which consists of the maximum of two affine functions. However, the proof also holds if more terms are present.

$$= \max\big(-\max(|\beta|f_1,|\beta|f_2), -\max(|\beta|f_3,|\beta|f_4)\big) \qquad \text{(by (8))}$$
$$= \max\big(\min(-|\beta|f_1,-|\beta|f_2), \min(-|\beta|f_3,-|\beta|f_4)\big) \qquad \text{(by (8))}$$
$$= \max\big(\min(\beta f_1,\beta f_2), \min(\beta f_3,\beta f_4)\big)$$
$$= \min\big(\max(\beta f_1,\beta f_3), \max(\beta f_1,\beta f_4), \max(\beta f_2,\beta f_3), \max(\beta f_2,\beta f_4)\big) \quad \text{(by (5))}. \quad \square$$

4 MPC for MMPS systems

In this section we give a short overview of the main results of [12] in which we have extended the MPC framework to MMPS systems. Related results can be found in [1, 4]. We assume that the reader is familiar with the basics of MPC, i.e., MPC is a model-based control approach that allows constraints on the inputs and outputs; in MPC at each sample step the optimal control inputs that minimize a given objective function over a given prediction horizon are computed, and applied using a receding horizon approach[4]. More information can be found in [7, 15, 22] and the references therein.

We can use the deterministic model (2) either as a model of an MMPS system, as the equivalent model of a continuous PWA system, or as an approximation of a general smooth nonlinear system. Note that we do not include modeling errors or uncertainty in the model. However, since MPC uses a receding finite horizon approach, we can regularly update the model and the current state as new information and new measurements become available. We also assume that the state is measurable[5].

In MMPS-MPC we compute at each sample step k an optimal control input that minimizes a cost criterion over the period $[k, k+N_p-1]$ where N_p is the prediction horizon. As we assume that at sample step k the current state can be measured, estimated or predicted using previous measurements, we can make an estimate $\hat{y}(k+j|k)$ of the output of the model (2) at sample step $k+j$ based on the state $x(k-1)$ and the future inputs $u(k+i)$, $i=0,\ldots,j$. Using successive substitution, we obtain an expression of the form $\hat{y}(k+j|k) = F_j(x(k-1), u(k), \ldots, u(k+j))$ for $j = 0,\ldots,N_p-1$. Clearly, $\hat{y}(k+j|k)$ is an MMPS function of $x(k-1), u(k), \ldots, u(k+j)$.

The cost criterion $J(k) = J_{\text{out}}(k) + \lambda J_{\text{in}}(k)$ used in MMPS-MPC reflects the reference tracking error (J_{out}) and the control effort (J_{in}), where λ is a nonnegative weight parameter. Let r denote the reference signal, and define the vectors

$$\tilde{u}(k) = \begin{bmatrix} u^T(k) & \ldots & u^T(k+N_p-1) \end{bmatrix}^T, \quad \tilde{y}(k) = \begin{bmatrix} \hat{y}^T(k|k) & \ldots & \hat{y}^T(k+N_p-1|k) \end{bmatrix}^T,$$
$$\tilde{r}(k) = \begin{bmatrix} r^T(k) & \ldots & r^T(k+N_p-1) \end{bmatrix}^T .$$

In practical situations, there will be constraints on the input and output signals (caused by limited capacity of buffers, limited transportation rates, saturation, etc.). In general, this is reflected in an MMPS constraint of the form $C_c(k, x(k-1), \tilde{u}(k), \tilde{y}(k)) \geqslant 0$.

In this paper we consider the following output and input cost functions[6]:

$$J_{\text{out},1}(k) = \|\tilde{y}(k) - \tilde{r}(k)\|_1,\; J_{\text{out},\infty}(k) = \|\tilde{y}(k) - \tilde{r}(k)\|_\infty,\; J_{\text{in},1}(k) = \|\tilde{u}(k)\|_1,\; J_{\text{in},\infty}(k) = \|\tilde{u}(k)\|_\infty.$$

[4]This means that after computation of the optimal control sequence $u(k), u(k+1), \ldots, u(k+N_c-1)$, only the first control sample $u(k)$ will be implemented, subsequently the horizon is shifted one sample; next, the model and the state are updated using new information from the measurements, and a new MPC optimization is performed for sample step $k+1$.

[5]Alternatively, one might assume that the state can be estimated. However, this is not a trivial operation as the system is operating in closed-loop and as — to the authors' best knowledge — observability and state estimation for (general) PWA systems is still an open issue.

[6]In conventional MPC usually quadratic cost functions of the form $J_{\text{out}}(k) = \|\tilde{y}(k) - \tilde{r}(k)\|_2^2$ and $J_{\text{in}}(k) = \|\tilde{u}(k)\|_2^2$ are used. In a discrete event context, however, other choices are more appropriate (see [11, 12]).

Note that these cost functions are also MMPS functions (recall that $|x| = \max(x, -x)$ for $x \in \mathbb{R}$).

The MMPS-MPC problem at sample step k consists in minimizing $J(k)$ over all possible future input sequences subject to the constraints. Just as in conventional MPC, a control horizon N_c is introduced in MPC, which means that the input is taken to be constant beyond sample step $k+N_c$:

$$u(k+j) = u(k+N_c-1) \quad \text{for } j = N_c, \ldots, N_p-1. \tag{11}$$

Alternatively, we can set the input rate constant as was done in [12]. In addition to a decrease in the number of optimization parameters and thus also the computational burden, a smaller control horizon N_c also gives a smoother control signal, which is often desired in practical situations. On the other hand, N_c should also not be too small since otherwise the controller may not have enough degrees of freedom to reach constraints and the control objectives.

5 Algorithms for the MMPS-MPC optimization problem

5.1 Nonlinear optimization

In general the MMPS-MPC optimization problem is a nonlinear, nonconvex optimization problem. In [12] we have discussed some algorithms to solve the MMPS-MPC optimization problem such as multi-start nonlinear optimization based on sequential quadratic programming (SQP), or a method based on the extended linear complementarity problem (ELCP). However, both methods have their disadvantages. If we use the SQP approach, then we usually have to consider a large number of initial starting points and perform several optimization runs to obtain (a good approximation to) the global minimum. In addition, the objective functions that appear in the MMPS-MPC optimization problem are non-differentiable and PWA (if we use the cost criteria $J_{\text{out},1}$, $J_{\text{out},\infty}$, $J_{\text{in},1}$, $J_{\text{in},\infty}$, or those given in [11]), which makes the SQP approach less suitable for them. The main disadvantage of the ELCP approach is that the execution time of this algorithm increases exponentially as the size of the problem increases. This implies that this approach is not feasible if N_c or the number of inputs and outputs of the system are large.

An alternative option consists in transforming the MMPS system into a mixed-logic (MLD) system [4] since MMPS systems are equivalent to MLD systems [17]. The main difference between MLD-MPC and MMPS-MPC is that MLD-MPC requires the solution of *mixed integer-real* optimization problems. In general, these are also computationally hard optimization problems. The on-line computational cost can be reduced by off-line computation [1], but this approach leads to high storage space requirements.

In the next section we will present another method to solve the MMPS-MPC optimization problem that is similar to the cutting-plane method used in convex optimization.

5.2 A new algorithm

We assume that the cost criteria $J_{\text{out},1}$, $J_{\text{out},\infty}$, $J_{\text{in},1}$, and/or $J_{\text{in},\infty}$ are used[7]. Recall that these objective functions (and any linear combination of them) are MMPS functions. The same holds for the estimate of future output $\tilde{y}(k)$. So if we substitute $\tilde{y}(k)$ in the expression for $J(k)$, we finally obtain an MMPS function of $\tilde{u}(k)$ as objective function. From Theorem 3.1 it follows that this objective function can

[7] The result below also holds for any other cost criterion that is an MMPS function of $\tilde{y}(k)$ and $\tilde{u}(k)$. It follows from Theorem 2.3 that any continuous PWA norm function can also be used.

be written in min-max canonical form as follows[8]:

$$J(k) = \min_{i=1,\dots,\ell} \max_{j=1,\dots,n_i} (\alpha_{(i,j)}^T(k)\tilde{u}(k) + \beta_{(i,j)}(k))$$

for appropriately defined integers ℓ, $n_1,\dots,n_\ell$, vectors $\alpha_{(i,j)}(k)$ and scalars $\beta_{(i,j)}(k)$. In general, the expression obtained by straightforwardly applying the manipulations of the proof of Theorem 3.1 may contain a large number of affine arguments $\alpha_{(i,j)}^T(k)\tilde{u}(k)+\beta_{(i,j)}(k)$. However, many of these terms are redundant[9] and can thus be removed. This reduces the number of affine arguments. Also note that the transformation into canonical form has to be performed only once — provided that we explicitly consider all arguments that depend on k as additional variables when performing the transformation, — and that it can be done off-line.

The derivation below is similar to the cutting-plane algorithm for convex optimization (see, e.g., [5]). Hence, it requires constraints that are linear (or convex) in $\tilde{u}(k)$. Note that the control horizon constraint (11) satisfies this condition. However, even if the original MPC constraint $C_c(k,x(k-1),\tilde{u}(k),\tilde{y}(k)) \geqslant 0$ is linear in $\tilde{u}(k)$ and $\tilde{y}(k)$, then in general this constraint is not linear any more after substitution of $\tilde{y}$. Therefore, from now on we assume that there are only linear constraints on the input:

$$P(k)\tilde{u}(k) + q(k) \geqslant 0 \ . \tag{12}$$

In practice such constraints occur if we have to guarantee that the control signal $\tilde{u}(k)$ or the control signal rate $\Delta\tilde{u}(k)$ stay within certain bounds. The optimization algorithm used below, which is based on the cutting plane algorithm for convex optimization, can also deal with convex constraints. So we can also allow convex constraints in $\tilde{u}(k)$ instead of (12). Furthermore, state or output constraints are allowed provided that after substitution they lead to a linear or convex constraint in $\tilde{u}(k)$.

To obtain the optimal MPC input signal at sample step k, we now have to solve an optimization problem of the following form:

$$\min_{\tilde{u}(k)} \min_{i=1,\dots,\ell} \max_{j=1,\dots,n_i} (\alpha_{(i,j)}^T(k)\tilde{u}(k) + \beta_{(i,j)}(k)) \qquad \text{subject to } P(k)\tilde{u}(k)+q(k) \geqslant 0 .$$

or equivalently

$$\min_{i=1,\dots,\ell} \min_{\tilde{u}(k)} \max_{j=1,\dots,n_i} (\alpha_{(i,j)}^T(k)\tilde{u}(k) + \beta_{(i,j)}(k)) \qquad \text{subject to } P(k)\tilde{u}(k)+q(k) \geqslant 0 \ . \tag{13}$$

Now let $i \in \{1,\dots,\ell\}$ and consider the subproblem

$$\min_{\tilde{u}(k)} \max_{j=1,\dots,n_i} (\alpha_{(i,j)}^T(k)\tilde{u}(k) + \beta_{(i,j)}(k)) \qquad \text{subject to } P(k)\tilde{u}(k)+q(k) \geqslant 0 \ ,$$

which is equivalent to the following linear programming (LP) problem:

$$\min_{t(k),\tilde{u}(k)} t(k) \quad \text{subject to} \begin{cases} t(k) \geqslant \alpha_{(i,j)}^T(k)\tilde{u}(k) + \beta_{(i,j)}(k) & \text{for } j=1,\dots,n_i \\ P(k)\tilde{u}(k)+q(k) \geqslant 0 . \end{cases} \tag{14}$$

This LP problem can be solved efficiently using a simplex method or an interior-point algorithm [23, 28]. To obtain the solution of (13), we solve (14) for $i=1,\dots,\ell$ and afterward we select the solution $\tilde{u}_{(i)}^{\text{opt}}(k)$ for which $\max\limits_{j=1,\dots,n_i} (\alpha_{(i,j)}^T(k)\tilde{u}_{(i)}^{\text{opt}}(k) + \beta_{(i,j)}(k))$ is the smallest. This results in an algorithm to solve the MMPS-MPC problem that is more efficient than the SQP or ELCP approach.

[8] Note that $\alpha_{(i,j)}$ and $\beta_{(i,j)}$ also depend on k via, e.g., the current state $x(k-1)$.

[9] E.g., since they appear twice, or since there are other arguments in the max (min) expression that are always larger (smaller) than the given argument.

Remark 5.1 The worst case complexity of the approach presented above is largely determined by the number of LPs to be solved, i.e., the number of linear terms in the equivalent min-max canonical form. In the worst case scenario this number increases very rapidly as the prediction horizon, the number of states of the MMPS systems, or the number of max-min nestings in the state equations or the objective function increases. However, although the number of terms in the full min-max canonical expression may be very large, it can sometimes be reduced significantly as will be illustrated in Example 6.2 (where the full canonical form contains 216 max-terms, of which only 4 are necessary). Although to the authors' best knowledge there are currently not yet any efficient algorithms for the simplification and reduction to a minimal canonical form (i.e., the canonical form with the minimal number of terms), some ad-hoc methods can be used to reduce the number of max-terms significantly (cf. Example 6.2). Furthermore, the complexity of the reduction process can also be reduced by already eliminating redundant terms during the intermediate steps of the transformations. Also note that this reduction may be done off-line. Furthermore, if we use a primal-dual simplex method or an interior-point method to solve the LP problems, we can improve the efficiency of the approach even further by stopping the optimization if we obtain a lower bound for the objective function of the current LP problem that is larger than the smallest final objective function of the LP problems that have already been solved.

6 Worked examples

In this section we discuss two examples. The first one is kept as simple as possible in order to show the basic ideas and all the intermediate steps taken. The second one is more complex. For both examples we provide a comparison with other methods.

Example 6.1 Consider the following PWA state space model:

$$y(k) = x(k) = \begin{cases} 0.5\,x(k-1) + 4\,u(k) - 1 & \text{if } 0.5\,x(k-1) + 3.8\,u(k) \leqslant 2 \\ 0.2\,u(k) + 1 & \text{if } 0.5\,x(k-1) + 3.8\,u(k) > 2\,. \end{cases} \tag{15}$$

It is easy to verify that the PWA function on the right-hand side of (15) is continuous[10]. Hence, (15) represents a continuous PWA system.

By applying the various properties of the max and min operators given in Section 3 of the main paper, or by applying the transformations given in the constructive proofs of [16, 24] (cf. Theorem 2.3), it is easy to verify that (15) is equivalent to the following MMPS system:

$$x(k) = \min(0.5\,x(k-1) + 4\,u(k) - 1, 0.2\,u(k) + 1) \tag{16}$$

$$y(k) = x(k) \;\; . \tag{17}$$

Let us now apply MPC to the system (16)–(17) using the optimization approach presented in Section 5.2. Suppose that we have the following constraints:

$$-0.2 \leqslant \Delta u(k) \leqslant 0.2 \quad \text{and} \quad u(k) \geqslant 0 \qquad \text{for all } k. \tag{18}$$

[10]If we rewrite (15) as $y(k) = x(k) = f(x(k-1), u(k))$, then we have — referring to Definition 2.2: $R_{(1)} = \{(x(k-1), u(k)) \in \mathbb{R}^2 \,|\, 0.5\,x(k-1) + 3.8\,u(k) \leqslant 2\}$, $R_{(2)} = \{(x(k-1), u(k)) \in \mathbb{R}^2 \,|\, 0.5\,x(k-1) + 3.8\,u(k) \geqslant 2\}$, $\alpha_{(1)} = \begin{bmatrix} 0.5 & 4 \end{bmatrix}^T$, $\beta_{(1)} = -1$, $\alpha_{(2)} = \begin{bmatrix} 0 & 0.2 \end{bmatrix}^T$, and $\beta_{(2)} = 1$. It is easy to verify that f is continuous on the boundary of $R_{(1)}$ and $R_{(2)}$.

Let[11] $N_c = N_p = 2$, and assume that the MPC objective function $J(k)$ is given by

$$J(k) = J_{\text{out},\infty}(k) + \lambda J_{\text{in},1}(k) = \|\tilde{y}(k) - \tilde{r}(k)\|_\infty + \lambda \|\tilde{u}(k)\|_1 \ ,$$

where $\lambda > 0$ is a weighting parameter and $r(k)$ the reference signal. Now we have

$$\begin{aligned}
J(k) &= \max\big(\,|y(k)-r(k)|, |y(k+1)-r(k+1)|\,\big) + \lambda\,\big(\,|u(k)|+|u(k+1)|\,\big) \\
&= \max\big(y(k)-r(k), r(k)-y(k), y(k+1)-r(k+1), r(k+1)-y(k+1)\big) + \\
&\quad \lambda\,\big(u(k)+u(k+1)\big) \qquad\qquad \text{(by (18))} \\
&= \max\big(y(k)-r(k)+\lambda u(k)+\lambda u(k+1), r(k)-y(k)+\lambda u(k)+\lambda u(k+1), \\
&\quad y(k+1)-r(k+1)+\lambda u(k)+\lambda u(k+1), r(k+1)-y(k+1)+\lambda u(k)+\lambda u(k+1)\big)\,.
\end{aligned} \tag{19}$$

By using successive substitution and by applying the properties given in Section 3 of the main paper, $y(k)$ and $y(k+1)$ can be expressed as functions of the current state $x(k-1)$ and the future inputs $u(k)$ and $u(k+1)$:

$$\begin{aligned}
y(k) = x(k) &= \min\big(0.5\,x(k-1)+4u(k)-1, 0.2\,u(k)+1\big) \\
y(k+1) = x(k+1) &= \min\big(0.5\,x(k)+4\,u(k+1)-1, 0.2\,u(k+1)+1\big) \\
&= \min\big(0.25\,x(k-1)+2\,u(k)+4\,u(k+1)-1.5, \\
&\qquad 0.1\,u(k)+4\,u(k+1)-0.5, 0.2\,u(k+1)+1\big) \ .
\end{aligned}$$

Using these expressions to eliminate $y(k)$ and $y(k+1)$ from the expression for $J(k)$ yields

$$\begin{aligned}
J(k) = \max\Big(&\min\big((\lambda+4)u(k)+\lambda u(k+1)-r(k)+0.5\,x(k-1)-1, \\
&\qquad (\lambda+0.2)u(k)+\lambda u(k+1)-r(k)+1\big), \\
&(\lambda-4)u(k)+\lambda u(k+1)+r(k)-0.5\,x(k-1)+1, \\
&(\lambda-0.2)u(k)+\lambda u(k+1)+r(k)-1\big), \\
&\min\big((\lambda+2)u(k)+(\lambda+4)u(k+1)-r(k+1)+0.25\,x(k-1)-1.5, \\
&\qquad (\lambda+0.1)u(k)+(\lambda+4)u(k+1)-r(k+1)-0.5, \\
&\qquad \lambda u(k)+(\lambda+0.2)u(k+1)-r(k+1)+1\big), \\
&(\lambda-2)u(k)+(\lambda-4)u(k+1)+r(k+1)-0.25\,x(k-1)+1.5, \\
&(\lambda-0.1)u(k)+(\lambda-4)u(k+1)+r(k+1)+0.5, \\
&\lambda u(k)+(\lambda-0.2)u(k+1)+r(k+1)-1\big)\Big) \ .
\end{aligned} \tag{20}$$

Note that this is an MMPS expression in max-min canonical form. In order to be able to apply the method of Section 5.2 we have to rewrite $J(k)$ into min-max canonical form. Equation (20) can be written compactly as[12]

$$J(k) = \max\big(\min(t_1,t_2), m_1, m_2, \min(t_3,t_4,t_5), m_3, m_4, m_5\big)$$

[11] We take such low values for N_c and N_p so that the size of the analytic expression for the MPC objective function $J(k)$ is still small so that it can be listed explicitly. In practice, larger values would be more appropriate, especially for more complex, higher-order systems.

[12] For the sake of simplicity of notation we will omit the arguments of the functions $t_1,\ldots,t_5$ and $m_1,\ldots,m_5$.

Method	CPU time (s)
LP	0.35
SQP	2.98
MILP	1.25
ELCP	12.13

Table 1: CPU time required for computing and simulating the closed-loop MPC input sequence over the period $[1,K]$ for Example 6.1 (average over 10 runs, with 2 significant decimal digits).

where $t_1,\ldots,t_5$ and $m_1,\ldots,m_5$ are appropriately defined affine functions of $x_1(k-1)$, $u(k)$, $u(k+1)$, and $r(k)$. The min-max canonical form of $J(k)$ is then given by

$$\begin{aligned} J(k) = \min\bigl(&\max(t_1,t_3,m_1,m_2,m_3,m_4,m_5),\max(t_1,t_4,m_1,m_2,m_3,m_4,m_5),\\ &\max(t_1,t_5,m_1,m_2,m_3,m_4,m_5),\max(t_2,t_3,m_1,m_2,m_3,m_4,m_5),\\ &\max(t_2,t_4,m_1,m_2,m_3,m_4,m_5),\max(t_2,t_5,m_1,m_2,m_3,m_4,m_5)\bigr) \ . \end{aligned} \tag{21}$$

Hence, the optimal MPC strategy for step k can be computed by solving six LP problems[13], and by selecting the overall optimum.

Let us now compute the closed-loop MPC input signal over a simulation period $[1,K]$ with $K=15$, $\lambda=0.05$, $x(0)=0.5$, $u(0)=0.1$, and for the reference signal r defined by

$$\{r(k)\}_{k=1}^{15} = 0.5,\ 0.8,\ 1,\ 1.5,\ 1.2,\ 1,\ 0.4,\ -0.5,\ -1.8,\ -1,\ -0.2,\ 0.8,\ 1,\ 1.1,\ 1 \ .$$

This results in the following closed-loop MPC input sequence[14]:

$$\begin{aligned} \{u_{\mathrm{mpc}}(k)\}_{k=1}^{15} = &\ 0.3,\ 0.394,\ 0.594,\ 0.794,\ 0.6,\ 0.4,\ 0.215,\ 0.015,\ 0,\ 0.171,\\ &\ 0.325,\ 0.475,\ 0.4,\ 0.5,\ 0.363 \ . \end{aligned}$$

In Figure 1 we have plotted the closed-loop MPC input signal u, the output signal y, the reference signal r, and the difference signal $y-r$.
We have solved the MPC optimization problem defined above for each sample step k using four different approaches:

1. the new LP based approach of Section 5.2,
2. a nonlinear nonconvex SQP approach,
3. the mixed integer linear programming (MILP) approach of [4],
4. the ELCP approach (cf. [12]).

The programs and functions to compute the optimal closed-loop MPC input sequence for each of the approaches above have been implemented in Matlab. For solving the LP, SQP, and MILP optimization problems we have respectively used the `linprog` function of the Matlab Optimization Toolbox [26], the `fmincon` function of the Matlab Optimization Toolbox, and the mixed integer linear and

[13] Each LP problem has 13 inequalities (7 coming from the objective function and 6 from the constraints (18) for k and $k+1$, and 2 variables ($u(k)$ and $u(k+1)$).

[14] The numerical values are given with 3 significant decimal digits.

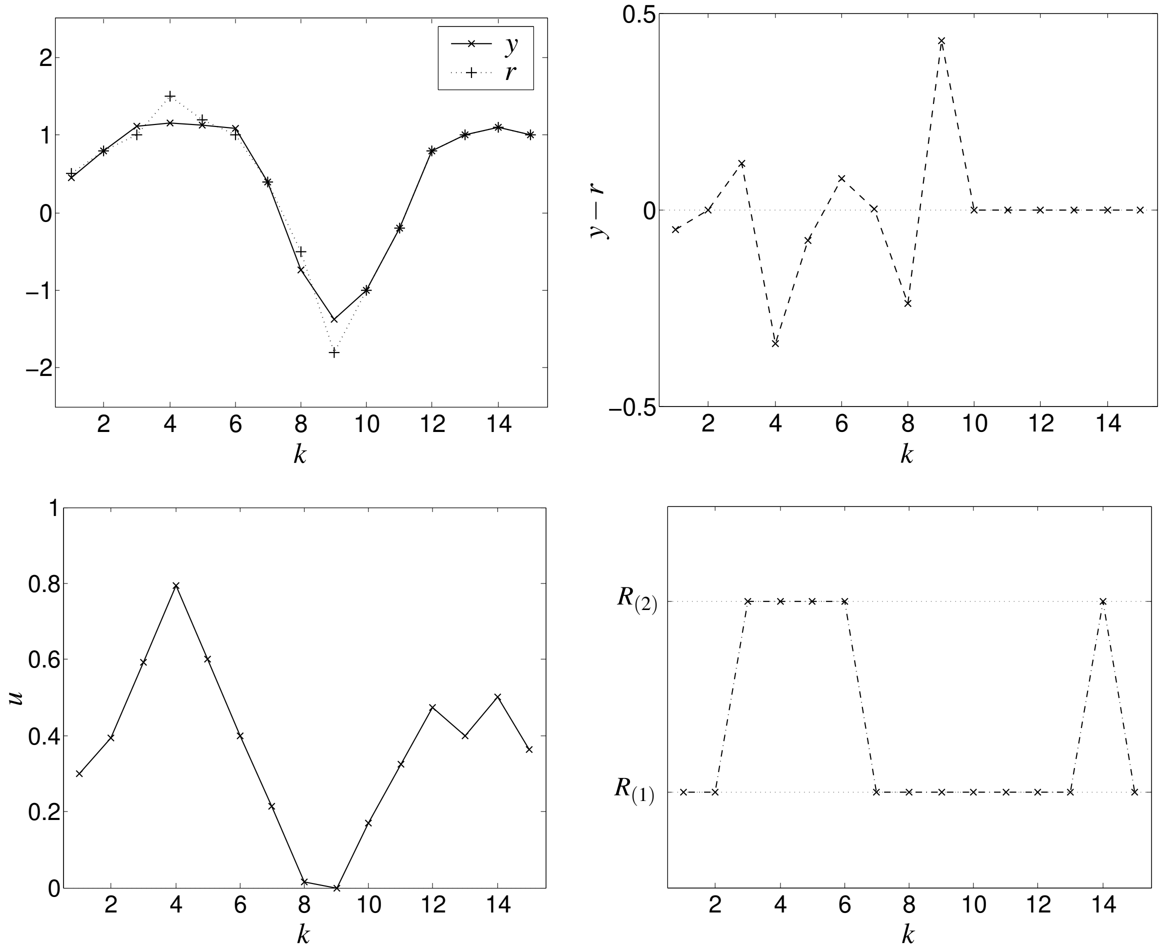

Figure 1: The closed-loop MPC output signal y, the reference signal r, the difference signal $y-r$, the input signal u, and the region $R_{(i)}$ (cf. Footnote 10) in which the system is at sample step k for Example 6.1.

quadratic programming function `miqp` [3] (in combination with `linprog`). Table 1 lists the CPU time required to compute the optimal closed-loop MPC input sequence (simulation time included) for the system (15) over the period $[1,K]$ on a 2.2 GHz Pentium 4 PC with 512 MB RAM. Clearly, for this example, the new LP approach outperforms the other approaches.

Example 6.2 Let us (re)consider the example presented in the paper [18], in which the equivalence between general (i.e., continuous or discontinuous) PWA systems and *constrained* MMPS systems was proved. In Example 1 of [18] the following hybrid system was considered[15]:

$$x(k) = \begin{cases} x(k-1)+u(k) & \text{if } x(k-1)+u(k) \leqslant 1 \\ 1 & \text{if } x(k-1)+u(k) > 1\,, \end{cases} \tag{22}$$

which represents an integrator with upper saturation. It is easy to verify that (22) represents a continuous PWA system.

[15]In order to get the same notation for the state equations as in the main paper, we have shifted the state by one sample, i.e., we have replaced $x(k+1)$ and $x(k)$ of [18] by $x(k)$ and $x(k-1)$ respectively. Note that this does not change the behavior of the system if the input is shifted accordingly.

In [18] this system was recast as the following max-plus-scaling[16] system:

$$x(k) = x(k-1) + u(k) - \max(0, x(k-1) + u(k) - 1) \ . \tag{23}$$

By inspection from (22), by applying the various properties of the max and min operators given in Section 3 of the main paper to expression (23), or by applying the transformations given in the constructive proofs of [16, 24] (cf. Theorem 2.3), it is easy to verify that (22) is equivalent to the following MMPS system:

$$x(k) = \min(x(k-1) + u(k), 1) \ . \tag{24}$$

In order to obtain a model of the form (2) we add the following output equation:

$$y(k) = x(k) \ . \tag{25}$$

Let us now apply MPC to the system (24)–(25) using the optimization approach presented in this paper. Suppose that we have the following constraints[17]:

$$-0.3 \leqslant \Delta u(k) \leqslant 0.3 \qquad \text{for all } k \tag{26}$$

$$u(k) + u(k+1) \leqslant 0 \qquad \text{for all } k \tag{27}$$

$$y(k) \geqslant r(k) \qquad \text{for all } k. \tag{28}$$

Let $N_\mathrm{c} = N_\mathrm{p} = 2$, and assume that the MPC objective function $J(k)$ is given by

$$J(k) = J_{\mathrm{out},\infty}(k) + \lambda J_{\mathrm{in},1}(k) \ ,$$

where $\lambda > 0$ is a weighting parameter and $r(k)$ the reference signal.

Using an approach that is similar to the approach taken in Example 6.1, we obtain an expression of the following form for the max-min canonical form of $J(k)$ (see [13]):

$$\begin{aligned} J(k) = \max\big(&\min(t_1,t_2), \min(t_3,t_4), \min(t_5,t_6), \\ &\min(t_7,t_8,t_9), \min(t_{10},t_{11},t_{12}), \min(t_{13},t_{14},t_{15})\big) \end{aligned} \tag{29}$$

where $t_1,\ldots,t_{15}$ are affine functions of $x_1(k-1)$, $u(k)$, $u(k+1)$, and $r(k)$. The min-max canonical form of $J(k)$ is then given by

$$\begin{aligned} J(k) = &\min\big(\max(t_1,t_3,t_5,t_7,t_{10},t_{13}), \max(t_1,t_3,t_5,t_7,t_{10},t_{14}), \max(t_1,t_3,t_5,t_7,t_{10},t_{15}), \\ &\max(t_1,t_3,t_5,t_7,t_{11},t_{13}), \max(t_1,t_3,t_15,t_7,t_{11},t_{14}), \max(t_1,t_3,t_5,t_7,t_{11},t_{15}), \\ &\ldots \\ &\max(t_2,t_4,t_6,t_9,t_{12},t_{13}), \max(t_2,t_4,t_6,t_9,t_{12},t_{14}), \max(t_2,t_4,t_6,t_9,t_{12},t_{15})\big) \ . \end{aligned} \tag{30}$$

Note that (30) contains a huge number of max-terms, viz. $2 \times 2 \times 2 \times 3 \times 3 \times 3 = 216$ max-terms. Using an iterative computation in which in each step one of the max-terms is removed and it is verified

[16] Max-plus-scaling functions and systems are defined analogously to MMPS functions and systems but without the min operator.

[17] It can be shown that after substitution of $y(k)$ the constraint $y(k) \geqslant r(k)$ can be recast as a system of linear constraints in $u(k)$ (see also [13]). This shows that our approach can also deal with constraints on the output or the state provided that after substitution they result in constraints that are linear (or convex) in the input $\tilde{u}(k)$.

Method	CPU time (s)
LP	0.31
SQP	2.10
MILP	1.04
ELCP	2.21

Table 2: CPU time required for computing and simulating the closed-loop MPC input sequence for Example 6.2 (average over 10 runs).

whether the resulting reduced min-max function is equivalent[18] to the original min-max function, it can be shown that most of these max-terms in are redundant and can thus be removed, which yields

$$\begin{aligned} J(k) = \min\big(&\max(t_1,t_3,t_5,t_7,t_{10},t_{13}), \max(t_1,t_3,t_5,t_9,t_{12},t_{15}), \max(t_2,t_4,t_6,t_8,t_{11},t_{14}), \\ &\max(t_2,t_4,t_6,t_9,t_{12},t_{15})\big) \enspace . \end{aligned} \tag{31}$$

This finally results in four LP problems that have to be solved in each MPC step.

Let us now compute the closed-loop MPC input signal over a simulation period $[1,15]$ with $\lambda = 0.1$, $x(0) = 1$, $u(0) = -0.1$, and for the reference signal

$$\begin{aligned} \{r(k)\}_{k=1}^{15} = {}& 1,\ 1,\ 0.7,\ 0.5,\ -0.45,\ -0.9,\ -1.2,\ -1.5,\ -1.4,\ -2.4, \\ & -2.5,\ -2.6,\ -2.6,\ -2.75,\ -2.75 \enspace . \end{aligned}$$

This results in the following closed-loop MPC input sequence:

$$\begin{aligned} \{u_{\rm mpc}(k)\}_{k=1}^{15} = {}& 0,\ 0,\ -0.3,\ -0.2,\ -0.5,\ -0.75,\ -0.45,\ -0.25,\ 0.05,\ -0.25, \\ & -0.55,\ -0.35,\ -0.05,\ -0.15,\ 0 \enspace . \end{aligned}$$

We have also solved the MPC optimization problems using the new approach, the SQP approach, the ELCP approach, and the MILP approach. Table 2 lists the CPU time required to compute the optimal closed-loop MPC input sequence (simulation time included) for the system (22) over the period $[1,15]$ using the Matlab Optimization Toolbox and `miqp` on a 2.2 GHz Pentium 4 PC with 512 MB RAM. So, for this example, the new LP approach also outperforms the other approaches.

7 Conclusion

First, we have shown that continuous PWA systems are equivalent to MMPS systems. This result is a refinement of previous results since it does not require the introduction of auxiliary variables or additional MMPS constraints. Next, we have considered MPC for continuous PWA and MMPS systems. In general, this leads to nonlinear, nonconvex optimization problems. We have presented a method that is based on canonical forms for MMPS functions and that is similar to the cutting-plane convex optimization algorithm to solve these optimization problems. More specifically, the approach consists in solving several LP problems and afterward selecting the solution that yields the

[18] As MMPS functions are PWA functions, it is easy to verify that the equivalence of two MMPS functions (of n variables) can be determined by selecting for each polyhedral regions $R_{(i)}$ (cf. Definition 2.2) corresponding to one of the functions $n+1$ linearly independent points, and by comparing the values of both functions in all these points. For the sake of efficiency, it is useful to consider points that are common to two or more regions such as, e.g., the vertices of the regions.

smallest objective function. This results in a method that is more efficient than just applying nonlinear optimization as was done in previous research.

Topics for future research include: a thorough investigation and comparison of the performance and the efficiency of the different optimization algorithms that have been considered in this paper and in [1, 4], investigation and characterization of the computational complexity of the transformation into the canonical form[19], investigation and characterization of the (average) number of LP problems and the number of inequalities they contain, and extension of our results to include modeling errors and noise in a stochastic or an ℓ_∞ framework.

Acknowledgment: Research partially funded by the Dutch Technology Foundation STW projects "Model predictive control for hybrid systems" (DMR.5675) and "Multi-agent control of large-scale hybrid systems" (DWV.6188), and by the European IST project "Modelling, Simulation and Control of Nonsmooth Dynamical Systems (SICONOS)" (IST-2001-37172)

References

[1] A. Bemporad, F. Borrelli, and M. Morari, "Optimal controllers for hybrid systems: Stability and piecewise linear explicit form," in *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, pp. 1190–1194, Dec. 2000.

[2] A. Bemporad, G. Ferrari-Trecate, and M. Morari, "Observability and controllability of piecewise affine and hybrid systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1864–1876, 2000.

[3] A. Bemporad and D. Mignone, "miqp.m: A Matlab function for solving Mixed Integer Quadratic Programs," Tech. rep. AUT00-22, Institut für Automatik, Swiss Federal Institute of Technology (ETHZ), Zürich, Switzerland, Oct. 2000.

[4] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, Mar. 1999.

[5] S.P. Boyd and C.H. Barratt, *Linear Controller Design: Limits of Performance*. Englewood Cliffs, New Jersey: Prentice Hall, 1991.

[6] R.E. Burkard and P. Butkovič, "Finding all essential terms of a characteristic maxpolynomial," *Discrete Applied Mathematics*, vol. 130, pp. 367–380, 2003.

[7] E.F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*. Berlin, Germany: Springer-Verlag, 1995.

[8] L.O. Chua and A.C. Deng, "Canonical piecewise-linear representation," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 1, pp. 101–111, Jan. 1988.

[9] L.O. Chua and R.L.P. Ying, "Canonical piecewise-linear analysis," *IEEE Transactions on Circuits and Systems*, vol. 30, no. 3, pp. 125–140, 1983.

[10] R.A. Cuninghame-Green and P.F.J. Meijer, "An algebra for piecewise-linear minimax problems," *Discrete Applied Mathematics*, vol. 2, pp. 267–294, 1980.

[11] B. De Schutter and T. van den Boom, "Model predictive control for max-plus-linear discrete event systems," *Automatica*, vol. 37, no. 7, pp. 1049–1056, July 2001.

[12] B. De Schutter and T.J.J. van den Boom, "Model predictive control for max-min-plus-scaling systems," in *Proceedings of the 2001 American Control Conference*, Arlington, Virginia, pp. 319–324, June 2001.

[13] B. De Schutter and T.J.J. van den Boom, "MPC for continuous piecewise-affine systems — Addendum," Tech. rep. CSE02-004a, Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands, Oct. 2003.

[19] Some indications on how to tackle this problem might be found in the literature on max-plus functions [6, 10].

[14] G. Ferrari-Trecate, F.A. Cuzzola, D. Mignone, and M. Morari, "Analysis and control with performance of piecewise affine and hybrid systems," in *Proceedings of the 2001 American Control Conference*, Arlington, Virginia, pp. 200–205, June 2001.

[15] C.E. García, D.M. Prett, and M. Morari, "Model predictive control: Theory and practice — A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, May 1989.

[16] V.V. Gorokhovik and O.I. Zorko, "Piecewise affine functions and polyhedral sets," *Optimization*, vol. 31, pp. 209–221, 1994.

[17] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, July 2001.

[18] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad, "On the equivalence of classes of hybrid dynamical models," in *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida, pp. 364–369, Dec. 2001.

[19] M. Johansson, *Piecewise Linear Control Systems*, vol. 284 of *Lecture Notes in Control and Information Sciences*. Berlin, Germany: Springer, 2003.

[20] T.A.M. Kevenaar and D.M.W. Leenaerts, "A comparison of piecewise-linear model descriptions," *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications*, vol. 39, no. 12, pp. 996–1004, Dec. 1992.

[21] D.M.W. Leenaerts and W.M.G. van Bokhoven, *Piecewise Linear Modeling and Analysis*. Boston: Kluwer Academic Publishers, 1998.

[22] J.M. Maciejowski, *Predictive Control with Constraints*. Harlow, England: Prentice Hall, 2002.

[23] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*. Philadelphia, Pennsylvania: SIAM, 1994.

[24] S. Ovchinnikov, "Max-min representation of piecewise linear functions," *Beiträge zur Algebra und Geometrie/Contributions to Algebra and Geometry*, vol. 43, no. 1, pp. 297–302, 2002.

[25] E.D. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 346–358, Apr. 1981.

[26] The MathWorks, Natick, Massachusetts, *Optimization Toolbox User's Guide – Version 2.1*, 2000.

[27] V.M. Veliov and M.I. Krastanov, "Controllability of piecewise linear systems," *Systems & Control Letters*, vol. 7, pp. 335–341, 1986.

[28] S.J. Wright, *Primal-Dual Interior Point Methods*. Philadelphia, Pennsylvania: SIAM, 1997.

MPC for Continuous Piecewise-Affine Systems — Addendum*

B. De Schutter and T.J.J. van den Boom

Delft Center for Systems and Control, Delft University of Technology
Mekelweg 2, 2628 CD Delft, The Netherlands
email: {b.deschutter,t.j.j.vandenboom}@dcsc.tudelft.nl

Abstract

This addendum contains some extra information in connection with the worked examples of Section 6 of the paper "MPC for continuous piecewise-affine systems" (by B. De Schutter and T.J.J. van den Boom, *Systems & Control Letters*, vol. 52, no. 3–4, pp. 179–192, July 2004). In particular, we give the explicit form of the optimization problems for each of the four solution approaches used in Section 6.

All references in this addendum that are not preceded by a capital letter A refer to sections, equations, etc. of the paper [A2].

Example 6.1 (continued) For each of the solution approaches considered in Section 6 we get the following explicit form for the optimization problems:

1. **the new LP based approach of Section 5.2**:
   In this case we have to solve the six LPs that correspond to (21) subject to[1]

$$-0.2 \leqslant u(k) - u(k-1) \leqslant 0.2 \tag{A.1}$$

$$-0.2 \leqslant u(k+1) - u(k) \leqslant 0.2 \tag{A.2}$$

$$u(k) \geqslant 0 \tag{A.3}$$

$$u(k+1) \geqslant 0 \ ; \tag{A.4}$$

2. **a nonlinear nonconvex SQP approach**:
   Here we have to solve

$$\min_{u(k),u(k+1)} \max\big(|y(k)-r(k)|, |y(k+1)-r(k+1)|\big) + \lambda\,\big(u(k)+u(k+1)\big)$$

   subject to

$$y(k) = \min\big(0.5\,x(k-1) + 4\,u(k) - 1,\, 0.2\,u(k)+1\big)$$

$$y(k+1) = \min\big(0.25\,x(k-1) + 2\,u(k) + 4\,u(k+1) - 1.5,\; 0.1\,u(k) + 4\,u(k+1) - 0.5,\, 0.2\,u(k+1)+1\big)$$

   and (A.1)–(A.4);

3. **the mixed integer linear programming (MILP) approach of [4]**:
   This results in[2]

$$\min_{\substack{u(k),u(k+1),\delta(k),\delta_{k+1},\\ z_1(k+1),z_2(k),z_2(k+1),t(k)}} t(k) + \lambda u(k) + \lambda u(k+1)$$

*Note that this addendum is not a part of the published journal paper [A2]. However, it is available as a separate technical report [13].

[1] I.e., (18) for k and $k+1$.

[2] See [4] for the way to transform a PWA model into a mixed logical dynamical model (MLD) (i.e., a system with both boolean and real state variables, with linear state and output equations, and with additional linear inequality constraints on the state variables). The piecewise linear objective function $J(k) = \max\big(|y(k)-r(k)|, |y(k+1)-r(k+1)|\big) + \lambda\,\big(u(k)+u(k+1)\big)$ has been transformed into a linear objective function by introducing an extra variable $t(k) = \max\big(|y(k)-r(k)|, |y(k)-r(k+1)|\big)$. The 5 other extra variables ($\delta(k)$, $\delta(k+1)$, $z_1(k+1) = x(k)\delta(k+1)$, $z_2(k) = u(k)\delta(k)$, $z_2(k+1) = u(k+1)\delta(k+1)$) originate from the transformation from PWA into MLD equations. Note that since the value of $x(k-1)$ is known at sample step k the term $0.5x(k-1)\delta(k)$ is in fact a linear term. As a consequence, the equation for $x(k)$ is linear in $x(k)$, $\delta(k)$, $z_2(k)$ and $u(k)$.

subject to

$$
\begin{aligned}
& t(k) \geqslant x(k) - r(k) \\
& t(k) \geqslant r(k) - x(k) \\
& t(k) \geqslant x(k+1) - r(k+1) \\
& t(k) \geqslant r(k+1) - x(k+1) \\
& x(k) = 0.5\, x(k-1)\delta(k) + 3.8\, z_2(k) - 2\,\delta(k) + 0.2\, u(k) + 1 \\
& x(k+1) = 0.5\, z_1(k+1) + 3.8\, z_2(k+1) - 2\,\delta(k+1) + 0.2\, u(k+1) + 1 \\
& \varepsilon - (M_f + \varepsilon)\delta(k) \leqslant 0.5\, x(k-1) + 3.8\, u(k) - 2 \leqslant M_f(1-\delta(k)) \\
& -M_u\delta(k) \leqslant z_2(k) \leqslant M_u\delta(k) \\
& u(k) - M_u(1-\delta(k)) \leqslant z_2(k) \leqslant u(k) + M_u(1-\delta(k)) \\
& \varepsilon - (M_f + \varepsilon)\delta(k+1) \leqslant 0.5\, x(k) + 3.8\, u(k+1) - 2 \leqslant M_f(1-\delta(k+1)) \\
& -M_x\delta(k+1) \leqslant z_1(k+1) \leqslant M_x\delta(k+1) \\
& x(k) - M_x(1-\delta(k+1)) \leqslant z_1(k+1) \leqslant x(k) + M_x(1-\delta(k+1)) \\
& -M_u\delta(k+1) \leqslant z_2(k+1) \leqslant M_u\delta(k+1) \\
& u(k+1) - M_u(1-\delta(k+1)) \leqslant z_2(k+1) \leqslant u(k+1) + M_u(1-\delta(k+1)) \\
& \delta(k), \delta(k+1) \in \{0, 1\} \\
& \text{and (A.1)–(A.4),}
\end{aligned}
$$

with ε a small positive number, and with M_x an upper bound[3] for $|x(k)|$ for all k, and M_u an upper bound for $|u(k)|$ for all k, and $M_f = 0.5\, M_x + 3.8\, M_u + 2$;

4. **the ELCP approach (cf. [12])**:
Here we have to solve the following optimization problem[4]:

$$
\min_{\nu} \max\big(|y(k,\nu) - r(k)|, |y(k+1,\nu) - r(k+1)|\big) + \lambda\left(u(k,\nu) + u(k+1,\nu)\right)
$$

where ν contains the parameters of the parameterized solution set of the ELCP given below (this solution set can be computed with the ELCP algorithm of [A1]), and where $y(k,\nu)$, $y(k+1,\nu)$, $u(k,\nu)$, $u(k+1,\nu)$ are respectively the $y(k)$, $y(k+1)$, $u(k)$, $u(k+1)$ that correspond to the parameter vector ν. The ELCP is given by[5]

$$
\begin{aligned}
& 0.5\, x(k-1) + 4\, u(k) - 1 - y(k) \geqslant 0 \\
& 0.2\, u(k) + 1 - y(k) \geqslant 0 \\
& \big(0.5\, x(k-1) + 4\, u(k) - 1 - y(k)\big) \cdot \big(0.2\, u(k) + 1 - y(k)\big) = 0 \\
& 0.25\, x(k-1) + 2\, u(k) + 4\, u(k+1) - 1.5 - y(k+1) \geqslant 0 \\
& 0.1\, u(k) + 4\, u(k+1) - 0.5 - y(k+1) \geqslant 0 \\
& 0.2\, u(k+1) + 1 - y(k+1) \geqslant 0 \\
& \big(0.25\, x(k-1) + 2\, u(k) + 4\, u(k+1) - 1.5 - y(k+1)\big) \cdot \\
& \qquad \big(0.1\, u(k) + 4\, u(k+1) - 0.5 - y(k+1)\big) \cdot \\
& \qquad\qquad \big(0.2\, u(k+1) + 1 - y(k+1)\big) = 0 \\
& \text{and (A.1)–(A.4).}
\end{aligned}
$$

---

[3] The upper bounds M_x and M_u could be determined based on physical insight or on operational constraints.

[4] This problem can be solved using an SQP approach.

[5] See [12] for more information on how this ELCP should be constructed.

Remark A.1 For the system (16)–(17) we can also allow output constraints of the form[6]

$$y(k) \geqslant r(k) \qquad \text{for all } k \ . \tag{A.5}$$

Indeed, for k and $k+1$ this constraint leads to

$$\begin{aligned} y(k) &= \min\bigl(0.5\,x(k-1) + 4\,u(k) - 1,\, 0.2\,u(k) + 1\bigr) \geqslant r(k) \\ y(k+1) &= \min\bigl(0.25\,x(k-1) + 2\,u(k) + 4\,u(k+1) - 1.5, \\ &\qquad 0.1\,u(k) + 4\,u(k+1) - 0.5,\, 0.2\,u(k+1) + 1\bigr) \geqslant r(k+1) \end{aligned}$$

or equivalently

$$0.5\,x(k-1) + 4\,u(k) - 1 \geqslant r(k) \tag{A.6}$$

$$0.2\,u(k) + 1 \geqslant r(k) \tag{A.7}$$

$$0.25\,x(k-1) + 2\,u(k) + 4\,u(k+1) - 1.5 \geqslant r(k+1) \tag{A.8}$$

$$0.1\,u(k) + 4\,u(k+1) - 0.5 \geqslant r(k+1) \tag{A.9}$$

$$0.2\,u(k+1) + 1 \geqslant r(k+1) \ . \tag{A.10}$$

Since these constraints are affine in $\tilde{u}(k) = \begin{bmatrix} u(k) & u(k+1) \end{bmatrix}^T$, the new optimization approach of Section 5.2 can still be applied[7]. This also holds for constraints of the form

$$x(k) \geqslant x_{\text{low}}(k) \quad \text{and} \quad y(k) \geqslant y_{\text{low}}(k) \qquad \text{for all } k$$

for lower bound signals x_{low} and y_{low}, or for any *nonnegative* linear combination of these constraints[8]. This shows that our approach can also deal with constraints on the output or the state provided that after substitution they result in constraints that are convex in the input $\tilde{u}(k)$.

Example 6.2 (continued) Recall that we have selected the following MPC objective function $J(k)$:

$$J(k) = J_{\text{out},\infty}(k) + \lambda J_{\text{in},1}(k) \ .$$

Using the system equations (24)–(25) and the constraints (26)–(28) we obtain

$$\begin{aligned} J(k) &= \max\bigl(|y(k) - r(k)|,\, |y(k+1) - r(k+1)|\bigr) + \lambda\bigl(|u(k)| + |u(k+1)|\bigr) \\ &= \max\bigl(y(k) - r(k),\, y(k+1) - r(k+1)\bigr) + \qquad\qquad \text{(by (28))} \\ &\qquad \lambda\bigl(\max(u(k), -u(k)) + \max(u(k+1), -u(k+1))\bigr) \\ &= \max\bigl(y(k) - r(k),\, y(k+1) - r(k+1)\bigr) + \\ &\qquad + \lambda \max\bigl(u(k) + u(k+1),\, u(k) - u(k+1),\, -u(k) + u(k+1),\, -u(k) - u(k+1)\bigr) \\ &= \max\bigl(y(k) - r(k),\, y(k+1) - r(k+1)\bigr) + \\ &\qquad + \lambda \max\bigl(u(k) - u(k+1),\, -u(k) + u(k+1),\, -u(k) - u(k+1)\bigr) \qquad \text{(by (27))} \\ &= \max\bigl(y(k) - r(k) + \lambda u(k) - \lambda u(k+1),\, y(k) - r(k) - \lambda u(k) + \lambda u(k+1), \end{aligned}$$

[6] Since the output saturates at $0.2u(k) + 1$, we will have to adapt the reference signal r if the constraint (A.5) is added, since otherwise the MPC problem will be infeasible for some values of k (cf. conditions (A.7) and (A.10)).

[7] If the constraint (A.5) is added, the terms $r(k) - y(k)$ and $r(k+1) - y(k+1)$ in expression (19) for the objective function become redundant. As a consequence, the terms $m_1, \ldots, m_5$ will disappear from (21), but the constraints (A.6)–(A.10) will be added. Hence, we still have 6 LPs with 13 inequalities and 2 variables.

[8] However, constraints of the form $x(k) \leqslant x_{\text{upp}}(k)$ or $y(k) \leqslant y_{\text{upp}}(k)$ for upper bound signals x_{upp} and y_{upp} lead to constraints that are not convex in $\tilde{u}(k)$. Hence, if such constraints are present, the new optimization approach of Section 5.2 cannot be applied.

$$
\begin{aligned}
&y(k)-r(k)-\lambda u(k)-\lambda u(k+1), y(k+1)-r(k+1)+\lambda u(k)-\lambda u(k+1),\\
&y(k+1)-r(k+1)-\lambda u(k)+\lambda u(k+1),\\
&y(k+1)-r(k+1)-\lambda u(k)-\lambda u(k+1)\big)\quad .
\end{aligned}
$$

By using successive substitution and by applying the properties given in Section 3 of the main paper, $y(k)$ and $y(k+1)$ can be expressed as functions of the current state $x(k-1)$ and the future inputs $u(k)$ and $u(k+1)$:

$$
\begin{aligned}
y(k) = x(k) &= \min\big(x(k-1)+u(k),\, 1\big)\\
y(k+1) = x(k+1) &= \min\big(x(k)+u(k+1),\, 1\big)\\
&= \min\big(\min\big(x(k-1)+u(k),\, 1\big)+u(k+1),\, 1\big)\\
&= \min\big(\min\big(x(k-1)+u(k)+u(k+1),\, 1+u(k+1)\big),\, 1\big)\\
&= \min\big(x(k-1)+u(k)+u(k+1),\, u(k+1)+1,\, 1\big)\quad .
\end{aligned}
$$

Hence,

$$
\begin{aligned}
J(k) = \max\big(&\min\big(x(k-1)+u(k),\, 1\big)-r(k)+\lambda u(k)-\lambda u(k+1),\\
&\min\big(x(k-1)+u(k),\, 1\big)-r(k)-\lambda u(k)+\lambda u(k+1),\\
&\min\big(x(k-1)+u(k),\, 1\big)-r(k)-\lambda u(k)-\lambda u(k+1),\\
&\min\big(x(k-1)+u(k)+u(k+1), u(k+1)+1,\, 1\big)-r(k+1)+\lambda u(k)-\lambda u(k+1),\\
&\min\big(x(k-1)+u(k)+u(k+1), u(k+1)+1,\, 1\big)-r(k+1)-\lambda u(k)+\lambda u(k+1),\\
&\min\big(x(k-1)+u(k)+u(k+1), u(k+1)+1,\, 1\big)-r(k+1)-\lambda u(k)-\lambda u(k+1)\big)\\
= \max\big(&\min\big(x(k-1)+u(k)-r(k)+\lambda u(k)-\lambda u(k+1),\\
&\qquad 1-r(k)+\lambda u(k)-\lambda u(k+1)\big),\\
&\min\big(x(k-1)+u(k)-r(k)-\lambda u(k)+\lambda u(k+1),\\
&\qquad 1-r(k)-\lambda u(k)+\lambda u(k+1)\big),\\
&\min\big(x(k-1)+u(k)-r(k)-\lambda u(k)-\lambda u(k+1),\\
&\qquad 1-r(k)-\lambda u(k)-\lambda u(k+1)\big),\\
&\min\big(x(k-1)+u(k)+u(k+1)-r(k+1)+\lambda u(k)-\lambda u(k+1),\\
&\qquad u(k+1)+1-r(k+1)+\lambda u(k)-\lambda u(k+1),\\
&\qquad 1-r(k+1)+\lambda u(k)-\lambda u(k+1)\big),\\
&\min\big(x(k-1)+u(k)+u(k+1)-r(k+1)-\lambda u(k)+\lambda u(k+1),\\
&\qquad u(k+1)+1-r(k+1)-\lambda u(k)+\lambda u(k+1),\\
&\qquad 1-r(k+1)-\lambda u(k)+\lambda u(k+1)\big),\\
&\min\big(x(k-1)+u(k)+u(k+1)-r(k+1)-\lambda u(k)-\lambda u(k+1),\\
&\qquad u(k+1)+1-r(k+1)-\lambda u(k)-\lambda u(k+1),\\
&\qquad 1-r(k+1)-\lambda u(k)-\lambda u(k+1)\big)\big)\\
= \max\big(&\min\big(x(k-1)+(\lambda+1)u(k)-\lambda u(k+1)-r(k),\\
&\qquad \lambda u(k)-\lambda u(k+1)-r(k)+1\big),\\
&\min\big(x(k-1)+(-\lambda+1)u(k)+\lambda u(k+1)-r(k),\\
&\qquad -\lambda u(k)+\lambda u(k+1)-r(k)+1\big),\\
&\min\big(x(k-1)+(-\lambda+1)u(k)-\lambda u(k+1)-r(k),
\end{aligned}
$$

$$
\begin{aligned}
&-\lambda u(k)-\lambda u(k+1)-r(k)+1\big),\\
\min\big(&x(k-1)+(\lambda+1)u(k)+(-\lambda+1)u(k+1)-r(k+1),\\
&\lambda u(k)+(-\lambda+1)u(k+1)-r(k+1)+1,\\
&\lambda u(k)-\lambda u(k+1)-r(k+1)+1\big),\\
\min\big(&x(k-1)+(-\lambda+1)u(k)+(\lambda+1)u(k+1)-r(k+1),\\
&-\lambda u(k)+(\lambda+1)u(k+1)-r(k+1)+1,\\
&-\lambda u(k)+\lambda u(k+1)-r(k+1)+1\big),\\
\min\big(&x(k-1)+(-\lambda+1)u(k)+(-\lambda+1)u(k+1)-r(k+1),\\
&-\lambda u(k)+(-\lambda+1)u(k+1)-r(k+1)+1,\\
&-\lambda u(k)-\lambda u(k+1)-r(k+1)+1\big)\big)\ .
\end{aligned} \tag{A.11}
$$

Note that this is an MMPS expression in max-min canonical form, which can be written compactly as (29).

Recall that we have considered the computation of the closed-loop MPC input signal over a simulation period $[1,15]$ with $\lambda=0.1$, $x(0)=1$, $u(0)=-0.1$, and for the reference signal

$$
\begin{aligned}
\{r(k)\}_{k=1}^{15}=\ &1,\,1,\,0.7,\,0.5,\,-0.45,\,-0.9,\,-1.2,\,-1.5,\,-1.4,\,-2.4,\\
&-2.5,\,-2.6,\,-2.6,\,-2.75,\,-2.75\ .
\end{aligned}
$$

This results in the following closed-loop MPC input sequence:

$$
\begin{aligned}
\{u_{\rm mpc}(k)\}_{k=1}^{15}=\ &0,\,0,\,-0.3,\,-0.2,\,-0.5,\,-0.75,\,-0.45,\,-0.25,\,0.05,\,-0.25,\\
&-0.55,\,-0.35,\,-0.05,\,-0.15,\,0\ .
\end{aligned}
$$

In Figure A.1 we have plotted the closed-loop MPC input signal u, the output signal y, the reference signal r, and the difference signal $y-r$.

Remark A.2 Note that the constraint (28) leads to

$$
\begin{aligned}
y(k)&=\min\big(x(k-1)+u(k),\,1\big)\geqslant r(k)\\
y(k+1)&=\min\big(x(k-1)+u(k)+u(k+1),\,u(k+1)+1,\,1\big)\geqslant r(k+1)
\end{aligned}
$$

or equivalently

$$x(k-1)+u(k)\geqslant r(k) \tag{A.12}$$
$$1\geqslant r(k) \tag{A.13}$$
$$x(k-1)+u(k)+u(k+1)\geqslant r(k+1) \tag{A.14}$$
$$u(k+1)+1\geqslant r(k+1) \tag{A.15}$$
$$1\geqslant r(k+1)\ . \tag{A.16}$$

We have solved the MPC optimization problems using 4 different approaches:

1. **the new LP based approach of Section 5.2**:
   In this case we have to solve the four LPs that correspond to (31) subject to[9]

$$-0.3\leqslant u(k)-u(k-1)\leqslant 0.3 \tag{A.17}$$
$$-0.3\leqslant u(k+1)-u(k)\leqslant 0.3 \tag{A.18}$$

[9] See (26), (27), and (A.12), (A.14)–(A.15).

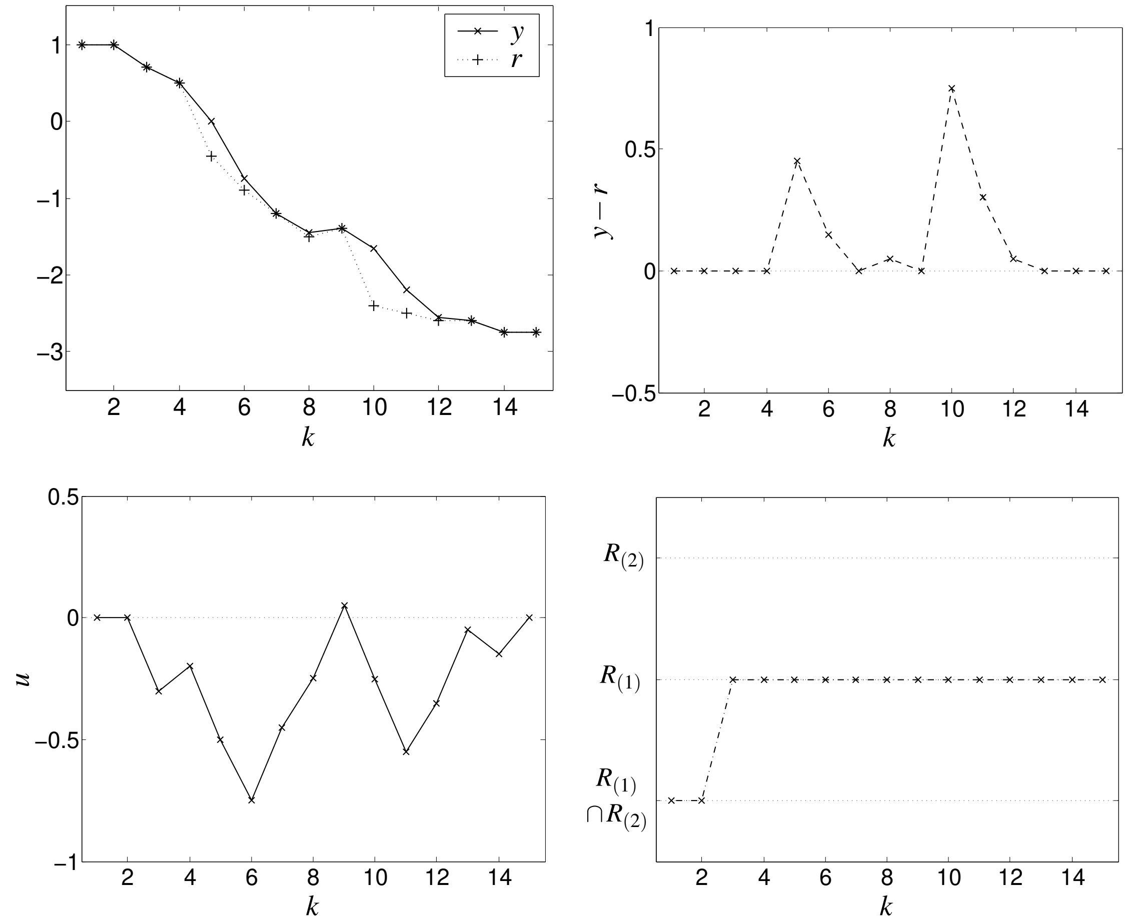

Figure A.1: The closed-loop MPC output signal y, the reference signal r, the difference signal $y-r$, the input signal u, and the region $R_{(i)}$ in which the system is at sample step k for Example 6.2.

$$u(k)+u(k-1) \leqslant 0 \quad \text{(A.19)}$$

$$u(k+1)+u(k) \leqslant 0 \quad \text{(A.20)}$$

$$x(k-1)+u(k) \geqslant r(k) \quad \text{(A.21)}$$

$$x(k-1)+u(k)+u(k+1) \geqslant r(k+1) \quad \text{(A.22)}$$

$$u(k+1)+1 \geqslant r(k+1) \quad ; \quad \text{(A.23)}$$

2. **a nonlinear nonconvex SQP approach**:
   Here we have to solve

$$\min_{u(k),u(k+1)} \max\big(y(k)-r(k), y(k+1)-r(k+1)\big) + \lambda\left(|u(k)|+|u(k+1)|\right)$$

subject to

$$y(k) = \min\big(x(k-1)+u(k),\, 1\big)$$

$$y(k+1) = \min\big(x(k-1)+u(k)+u(k+1),\, u(k+1)+1,\, 1\big)$$

$$y(k) \geqslant r(k)$$

$$y(k+1) \geqslant r(k+1)$$

and (A.17)–(A.20);

3. **the mixed integer linear programming approach of [4]**:
This results in[10]

$$\min_{\substack{u(k),u(k+1),\delta(k),\delta_{k+1},\\ z_1(k+1),z_2(k),z_2(k+1),t_1(k),t_2(k),t_3(k)}} t_1(k)+\lambda t_2(k)+\lambda t_3(k)$$

subject to

$$\begin{aligned}
&t_1(k) \geqslant x(k)-r(k)\\
&t_1(k) \geqslant x(k+1)-r(k+1)\\
&t_2(k) \geqslant u(k)\\
&t_2(k) \geqslant -u(k)\\
&t_3(k) \geqslant u(k+1)\\
&t_3(k) \geqslant -u(k+1)\\
&x(k) = x(k-1)\delta(k)+z_2(k)-\delta(k)+1\\
&x(k+1) = z_1(k+1)+z_2(k+1)-\delta(k+1)+1\\
&\varepsilon-(M_x+M_u+1+\varepsilon)\delta(k) \leqslant x(k-1)+u(k)-1 \leqslant (M_x+M_u+1)(1-\delta(k))\\
&-M_u\delta(k) \leqslant z_2(k) \leqslant M_u\delta(k)\\
&u(k)-M_u(1-\delta(k)) \leqslant z_2(k) \leqslant u(k)+M_u(1-\delta(k))\\
&\varepsilon-(M_x+M_u+1+\varepsilon)\delta(k+1) \leqslant x(k)+u(k+1)-1\\
&\qquad\qquad\qquad\qquad \leqslant (M_x+M_u+1)(1-\delta(k+1))\\
&-M_x\delta(k+1) \leqslant z_1(k+1) \leqslant M_x\delta(k+1)\\
&x(k)-M_x(1-\delta(k+1)) \leqslant z_1(k+1) \leqslant x(k)+M_x(1-\delta(k+1))\\
&-M_u\delta(k+1) \leqslant z_2(k+1) \leqslant M_u\delta(k+1)\\
&u(k+1)-M_u(1-\delta(k+1)) \leqslant z_2(k+1) \leqslant u(k+1)+M_u(1-\delta(k+1))\\
&x(k) \geqslant r(k)\\
&x(k+1) \geqslant r(k+1)\\
&\delta(k),\delta(k+1) \in \{0,1\}
\end{aligned}$$

and (A.17)–(A.20),

with ε a small positive number, and with M_x an upper bound for $|x(k)|$ for all k, and M_u an upper bound for $|u(k)|$ for all k;

4. **the ELCP approach (cf. [12])**:
Here we have the following optimization problem:

$$\min_{\nu} \max\bigl(y(k,\nu)-r(k), y(k+1,\nu)-r(k+1)\bigr)+\lambda\left(|u(k,\nu)|+|u(k+1,\nu)|\right)$$

where ν contains the parameters of the parameterized solution set of the ELCP given below as it can be computed with the ELCP algorithm of [A1] and $y(k,\nu)$, $y(k+1,\nu)$, $u(k,\nu)$, $u(k+1,\nu)$ respectively the $y(k)$, $y(k+1)$, $u(k)$, $u(k+1)$ that correspond to the parameter vector ν. The ELCP is given by

$$x(k-1)+u(k)-y(k) \geqslant 0$$

[10] See [4] for the way to transform a PWA model into an MLD model. The piecewise linear objective function $J(k) = \max\bigl(y(k)-r(k), y(k+1)-r(k+1)\bigr)+\lambda\left(|u(k)|+|u(k+1)|\right)$ has been transformed into a linear objective function by introducing 3 extra variables $t_1(k) = \max\bigl(y(k)-r(k), y(k)-r(k+1)\bigr)$, $t_2(k) = |u(k)| = \max\bigl(u(k), -u(k)\bigr)$, and $t_3(k) = |u(k+1)| = \max\bigl(u(k+1), -u(k+1)\bigr)$; the six other extra variables ($\delta(k)$, $\delta(k+1)$, $z_1(k)$, $z_1(k+1)$, $z_2(k)$, $z_2(k+1)$) originate from the transformation from PWA into MLD equations.

$$
\begin{aligned}
&1 - y(k) \geqslant 0 \\
&\big(x(k-1) + u(k) - y(k)\big) \cdot \big(1 - y(k)\big) = 0 \\
&x(k-1) + u(k) + u(k+1) - y(k+1) \geqslant 0 \\
&u(k+1) + 1 - y(k+1) \geqslant 0 \\
&1 - y(k+1) \geqslant 0 \\
&\big(x(k-1) + u(k) + u(k+1) - y(k+1)\big) \cdot \big(u(k+1) + 1 - y(k+1)\big) \cdot \\
&\qquad\qquad \big(1 - y(k+1)\big) = 0 \\
&y(k) \geqslant r(k) \\
&y(k+1) \geqslant r(k+1)
\end{aligned}
$$

and (A.17)–(A.20).

Additional references

[A1] B. De Schutter and B. De Moor, "The extended linear complementarity problem," *Mathematical Programming*, vol. 71, no. 3, pp. 289–325, Dec. 1995.

[A2] B. De Schutter and T.J.J. van den Boom, "MPC for continuous piecewise-affine systems," *Systems & Control Letters*, vol. 52, no. 3–4, July 2004.